

AFRL-VA-WP-TP-2002-310

**HUMAN/SYSTEM INTERFACE (HSI)
ISSUES FROM A USAF
PILOT/COMBAT UNMANNED AIR
VEHICLE (UAV) TEST OPERATOR
PERSPECTIVE**



Major Mark E. Garner

APRIL 2002

Approved for public release; distribution is unlimited.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

**AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542**

20020912 093

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YY) April 2002		2. REPORT TYPE Conference Paper Preprint		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE HUMAN/SYSTEM INTERFACE (HSI) ISSUES FROM A USAF PILOT/COMBAT UNMANNED AIR VEHICLE (UAV) TEST OPERATOR PERSPECTIVE				5a. CONTRACT NUMBER IN-HOUSE	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER N/A	
6. AUTHOR(S) Major Mark E. Garner				5d. PROJECT NUMBER N/A	
				5e. TASK NUMBER N/A	
				5f. WORK UNIT NUMBER N/A	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Control Sciences Division (VAC) Air Vehicles Directorate Air Force Research Laboratory, Air Force Materiel Command Wright-Patterson Air Force Base, OH 45433-7542				8. PERFORMING ORGANIZATION REPORT NUMBER AFRL-VA-WP-TP-2002-310	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AIR VEHICLES DIRECTORATE AIR FORCE RESEARCH LABORATORY AIR FORCE MATERIEL COMMAND WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542				10. SPONSORING/MONITORING AGENCY ACRONYM(S) AFRL/VAC	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-VA-WP-TP-2002-310	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES Paper presented at the AIAA 1st UAV Conference in Portsmouth, VA on 20-23 May 2002 and printed in the conference proceedings. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT (Maximum 200 Words) The complexity issues in dealing with the Human/System Interface (HSI) for an Unmanned Combat Air Vehicle (UCAV) system are numerous, varied, and difficult to overcome. I have been a part of the design and flight test team for one such system since early in the program. From this experience, I have come to conclusions that should help to reduce time and money spent on producing a useable, safe and robust interface for an UCAV system.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT: SAR	18. NUMBER OF PAGES 12	19a. NAME OF RESPONSIBLE PERSON (Monitor) Major Mark E. Garner 19b. TELEPHONE NUMBER (Include Area Code) (937) 255-4078
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			

HSI ISSUES FROM A USAF PILOT/COMBAT UAV TEST OPERATOR PERSPECTIVE

by Major Mark E. Garner, USAF
Dayton, Ohio

ABSTRACT

The complexity issues in dealing with the Human/System Interface (HSI) for an Unmanned Combat Air Vehicle (UCAV) system are numerous, varied and difficult to overcome. I have been a part of the design and flight test team for one such system since early in the program. From this experience, I have come to the following conclusions that should help to reduce time and money spent on producing a useable, safe and robust interface for an UCAV system:

- 1) Operators should be included in the design process from day one.
- 2) The design process should allow for immediate changes in the HSI
- 3) The HSI should be integrated with a highly accurate simulation to allow the operator to use in the interface in as real a scenario as possible as early in the design process as possible.
- 4) The HSI should be user-configurable and allow for multiple modes of feedback and input.

Bottom line: Design and implementation tools that allow real-time prototyping and use with a simulation are essential

THE PAPER

The first question I want to answer is why I am qualified to write a paper on such a complex issue. What follows is an anecdotal narrative about my experience as an Air Force strategic airlift pilot who became one person of hundreds working on an Advanced Technology Demonstration (ATD), the X-45 Unmanned Combat Air Vehicle, to prove that an unmanned bomber could perform Suppression of Enemy Air Defenses (SEAD).

From 1986 until 1991, I flew C-141B aircraft all over the world but spent a significant amount of time in and around Charleston Air force Base, South Carolina, training to fly 500 feet off of the ground and throw

people or cargo out of my aircraft. As a copilot I was certified for airdrop operations and quickly became a commodity for aircraft commanders who were taking their Lead Airdrop Aircraft Commander check ride. I was not necessarily an awesome pilot with "golden hands," but, I could run the cockpit for a pilot who was otherwise concerned with leading up to seven other aircraft around a visual flight rules (VFR) flight plan to get them to a drop zone on time and on target. My duties as the Lead Copilot included: talking on the radios to Air Traffic Control and the other aircraft; reading the map and putting waypoints into the navigation system; running checklists; monitoring the aircraft systems; and ensuring the Lead Aircraft Commander was not doing anything unsafe. My epiphany when I first heard a briefing about my unmanned bomber program was that all of the things that I did as a Lead Airdrop Copilot would also have to be done by the system or the system operator in order to get weapons to the target safely and on time. I immediately asked the program managers if I could become one of the initial operators for this system.

Both the Contractor and Government program managers agreed that I could be helpful to the program. I was asked to become part of several working groups including: Human/System Interface (HSI), Flight Test and Training, and Contingency Management. I have been most active on the HSI group and have recently participated in several flight operations and helped to integrate the mission control software and simulation into other Air Force simulations. What follows are challenges and opportunities that I have experienced during this work, strictly from my point-of-view. This material is not meant to be critical of anything or anyone specifically mentioned, it is just my opinion. Also, please realize the program I am working on is a design- in-progress and just because I have seen problems at this time does not mean those problems will not be resolved before this system is fielded. Most of the control software that I have been testing is from the first version of five more to follow before the system becomes operational. After I discuss some of the challenges I have observed, I will talk about my view of an optimum design system that could help any group trying to build a useable, safe and robust interface for an unmanned vehicle system.

The complexity issues in dealing with the HSI for an unmanned aircraft system are numerous, varied and difficult to overcome. Many initial questions such as the following must be answered before any design work

can take place: What information does the operator need? How should it be displayed? What controls does the operator need and how should they be displayed? Are communication rates going to affect the refresh rate of your displayed data? What can be done to alleviate operator load? How will feedback from the vehicle to the operator be displayed? How does the system alert the operator to a dangerous situation? Are there outside organization (i.e. FAA) requirements that must be met by the operator? The point of this paper is not to answer all of these questions as they pertain to the X-45, but to give a sense of how complex the HSI design issue can be.

Before any of the above questions can be answered, we need to know what our system will be doing and who will be operating this system. I will be using the term, target operators, which I mean to be those people with the required skills and experience to be operators for a particular unmanned system. Target operators should be included on the design team from day one. This is the first essential rule in HSI design. It seems trivial to have to mention this rule but if you look at some of the already-fielded systems, you will see that is not always the case. I have had the opportunity to use both the Predator and Global Hawk interfaces in a simulated environment and neither seem user-friendly to me. I can neither prove nor disprove that target operators were used in the design process for their HSI. However, documented problems with both systems have caused loss of an air vehicle.

Choosing a target operator for a design team is not a trivial matter either. Although the X-45 program has an idea who the operator on the fielded system will be, that idea is not written in stone. Air Combat Command has already acceded that preferred operator skills would be determined as the system evolves. According to **AIR COMBAT COMMAND CONCEPT OF OPERATIONS FOR ENDURANCE UNMANNED AERIAL VEHICLES, 3 Dec 1996 - Version 2, 8.2.1 Air Vehicle Operator (AVO)**, "Future AF endurance AVO qualification requirements will be determined during the demonstration phase of the ACTDs (Advanced Concept Technology Demonstrations). Once qualification requirements are determined, a detailed screening process and training syllabi will be developed. Potential AVOs may include AF enlisted members (new recruits or cross-trainees) who would undergo a carefully designed screening process to measure their potential to complete AVO training...The screening process will include academic, mechanical, medical, and physical skill testing. After successful screening, AVO candidates progress to the Undergraduate UAV Training Course. The

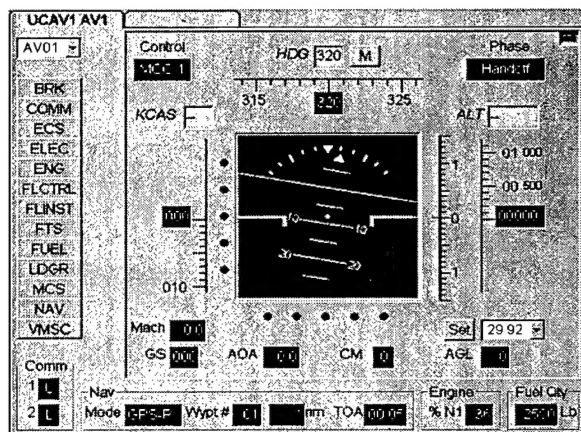
Undergraduate UAV Training Course will be similar to aviation ground school to include weather, flight planning, airmanship, radio procedures, and various other applicable disciplines. Upon completion of Undergraduate UAV training, candidates would proceed to a flight phase where the student learns basic flight principles." The Air Force Research Lab's Human Effectiveness Directorate is also working at this time to design some of these measurement processes and skills that will help determine who can be an unmanned bomber operator and how those operators should be trained.

There may be outside agencies that have requirements on these operators as well. What requirements does the Federal Aviation Administration (FAA) have on unmanned air vehicle operators? I have talked with at least one Global Hawk pilot that told me he has to be Instrument Flight Rule (IFR) certified, current and qualified in order to fly in federal airspace. Because he has been medically disqualified from flying, the Air Force pays for him to fly in Air Force Flying Club aircraft in order for him to maintain his IFR qualification. However, in 1996 the IFR qualification was not yet the requirement with the FAA. According to the same paper I cited earlier from Air Combat Command, "In addition, close coordination with the Federal Aviation Administration (FAA) will be required to ensure proper certification/training to fly in the National and International Airspace Systems. FAA certification of AVOs are not required at this time." As you can see, unmanned air vehicle operator requirements must continue to change and evolve.

It seems imperative to me that we work closely with the FAA to overcome the challenge of requiring IFR pilots to be UAV operators. The FAA has certified several simulators and companies to award pilots their instrument rating solely through simulated flight training. Perhaps before we become operational with the X-45 we can get certification on our simulation that would allow us to be current and certified as instrument UAV operators. This action would allow the unmanned vehicle program to realize its full benefit in life cycle cost savings. Since Air Force trained IFR pilots cost more than \$1 million dollars with all the actual flight time involved, training and certifying UAV operators in a simulation would lower training costs significantly.

Now I would like to discuss some of the significant challenges and their outcomes that I have observed from three years on my program. One of the longer discussions we had on our team was over the common pilot instruments such as airspeed, altimeter, attitude, direction and situation displays. Initially, since our

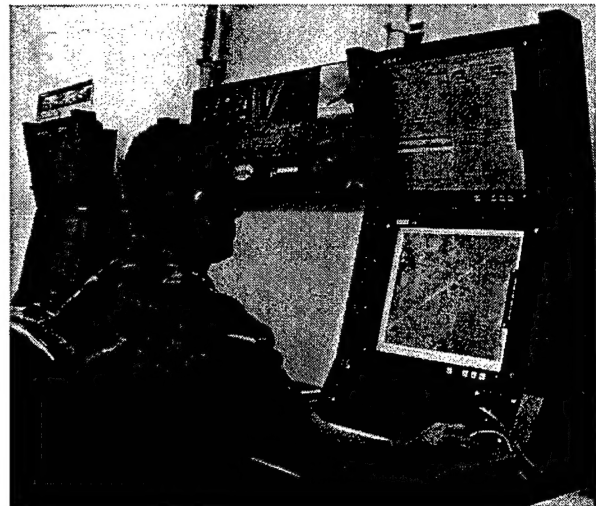
system would be totally automated so far as hands-on flying, I argued that we did not need the attitude display. Especially when monitoring more than one UCAV, an operator cannot be concerned with an individual vehicle's attitude. He will probably focus on the tactical or "God's eye view" display. Ultimately, I lost this argument since an attitude display is in our interface. In retrospect, I see how this display is necessary. Because we would only be flying one vehicle for the first part of our program during flight test, we would want as many eyes on the vehicle's performance as possible. Therefore the attitude display used by the operator would be very useful in providing that extra set of eyes. So where should this attitude display reside and how big should this display be?



Attitude indicator for X-45

In my program, the display area available to the operator is two 1280x1024 pixels, 20-inch diagonal, flat-screen liquid crystal displays. Considering most computer displays are configured for somewhere between 800x600 or 1000x800 pixels, our Mission Control Console (MCC) represents a huge amount of display real estate. We are using a point and click graphic user interface (GUI) for most commands in our system with one mouse and associated cursor. Originally, our cursor was a red arrow about the size of 10X5 pixels. Our cursor moves over background maps that have many colors on them, including red. These aspects working together make it very easy to lose the cursor on that huge display area. Many times during a simulation, I lost where the cursor was displayed. In fact, I could not figure out on which of the two screens it was displayed. This forced me to rapidly move the mouse back and forth to figure out where the cursor was displayed. This defect could cause numerous

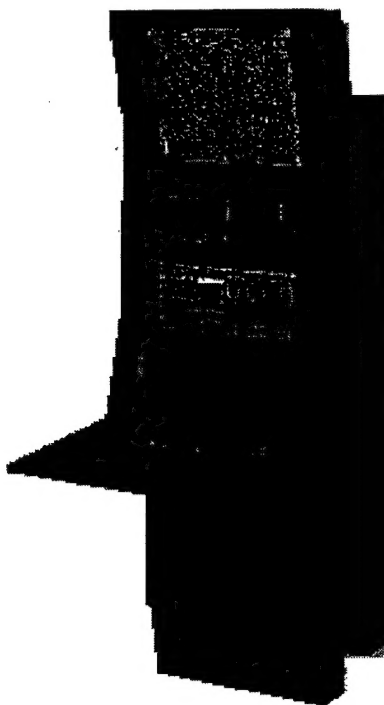
problems especially when the operator needs to get to a certain control and apply it quickly. So far, the solution for this challenge has been to make the cursor 4 times as big and to make the color a bright white. It is much easier to find the cursor on the displays now.



X-45 Mission Control Console

This story brings up another issue. How do we display commands for the operator to send to the vehicle. We are using several different taxonomies in my program. You can hook the vehicle icon with the cursor using the mouse and bring up a series of nested menus, click on buttons, or hook other icons on the tactical display to send commands. Certain phases of flight have a fly-in menu with clickable buttons. One of the taxonomies I do not like is the nested menu list. Some of the commands we can send are similarly named making it easy to send the wrong command. Some of these commands are time-sensitive as well, increasing the chance of hooking the wrong command if the operator is in a hurry or stressed. For example, on many occasions during demonstrations I have selected ATTACK versus DIRECT ATTACK when I meant to select DIRECT ATTACK. In this particular case, the difference in commands is fairly trivial but it does affect the simulation and in a real situation could subject the vehicle to a threat for a longer period. Although we have a way of canceling the command, there must be a better way of performing these operations. At some point, my program plans to implement some form of voice-activated control. Spoken commands could be used during the attack phase for many of the nested-menu, tactical commands.

Also, since we have a keyboard, I believe keyboard shortcuts would make the job easier provided certain rules are followed. It is a fact that a Predator was lost during training because of nested menus and keyboard shortcuts. Keyboard commands were context sensitive based upon the active window. For a hypothetical example, an "A" typed with the Takeoff window active would have meant Abort Takeoff. An "A" typed with the Tactical window active would have meant Attack. In the Predator's case, several windows were open and the operator thought he was doing something innocuous when in fact he was shutting down the engine. My rule for keyboard shortcuts: A typed shortcut would only mean one thing throughout the system, regardless of vehicle state. In other words, typing the Control key and the "C" key at the same time would always mean that weapons release consent has been activated. Typing that key during Takeoff phase would still mean the same thing, but it would not be applied to the system because the air vehicle is not in attack phase.



Another view of the MCC

Another challenge we have had with our display is the background map. Because we are also monitoring an autonomous taxi, we want to have a high degree of accuracy in our ground tactical display. Our program is using readily available digital maps from the Defense Mapping Agency that only zoom in to a 1-inch equals

2000 feet scale. When the map is at this scale our air vehicle icon covers the runway with its wings. Given that our wingspan is only about 30 feet wide, covering a 300 foot wide runway with its wings shows that we do not have the degree of fidelity we need to adequately monitor the vehicles progress on the runway, much less a 90 foot wide taxiway. A nose camera with a real-time video feed alleviates this problem, to a degree.

Another mitigating factor to this challenge is the addition of black background maps that allow the operator to zoom in further on the vehicle path. Waypoints on the ground taxi path that originally overlapped each other can now be separated by some black space. We can zoom in to one-inch equals one thousand feet. During taxi tests this allowed us to see the vehicle icon move when we commanded a 10-foot offset to the taxi path. We could see the vehicle icon move left or right away from the depicted taxi path. However, with the advent of detailed satellite imagery, using a geo-registered image of the airport area would be the best choice for a background image. A colorized version would be even better.

Early on, in my program's development, we ran into a semantic difficulty. The engineer that was designing the guidance and control algorithms for the air vehicle used the term, "glideslope intercept." In his algorithm the glideslope intercept point was the imaginary point that hit the runway if you drew a straight line from the air vehicle to the runway using a normal glideslope. To any instrument pilot, the glideslope intercept is something entirely different. It is the point that the pilot's aircraft intercepts the glideslope and begins its descent to the runway. I raised this point during design briefings and my argument was well received and understood. I was assured that the engineer's definition would never make its way to an operator's manual. Well, guess what? The engineer's definition was used in the first version of the manual. However, that definition has since been changed in subsequent editions.

One of the best demonstrations of how difficult the UAV operator's job could become was a taxi demonstration at Sunnyvale, California using the National Aeronautics and Space Administration's Future Flight Central (FFC) Simulator. The demo had a mock-up of the X-45 system's control station. It used proprietary control software for FFC that had a tactical display showing all air vehicles moving on the airport and another display showing simulated nose camera video. My control of the vehicle/vehicles consisted of START, STOP, FAST, SLOW and I could change the taxi/flight path at given intersections. I would control a

lead air vehicle and the other air vehicles would follow. Each vehicle took off singly and then rejoined after take-off. There was a real air-traffic controller and other pseudo-pilots to control the other aircraft. Everyone used the same ground/tower/Air Traffic Control frequency. I felt like I was actually in a cockpit trying to manage my aircraft. I had to maintain my situational awareness using my displays and listening to the radio. Displays consisted of a "God's eye view" tactical display where I could send commands to the vehicle and change its taxi path and a simulated nose-camera view. I became easily task saturated when the planned taxi/flight route changed in the canned flight plan. My cross check stayed mostly upon the nose camera view when my vehicles were on the ground. The demonstration allowed the contractor to gather data on operator workload when the air vehicles were operating out of a busy, wartime airfield with multiple types of aircraft in operation at the same time. I believe we have nose camera video on our vehicles because all of the test operators said the same thing; that **they relied upon the nose camera video for situational awareness.**

My reason for relaying these stories about the X-45 program is to help set in your mind just how very complex making an HSI for an unmanned vehicle can be. I have not tried to answer all the questions I brought up at the beginning of this paper. Even if I did, there would still be an infinite number of questions that have not yet been asked. So, given that the design process is terribly complex, how can we make it easier to get the HSI perfected and speed up the process at the same time? Well, we need a "perfect" HSI development environment.

Some of the key aspects of the perfect HSI development environment would be: Give the HSI engineer and the target operator access to a high quality simulation on which the operator can try out the prototype interface; Make the prototype interface reconfigurable for each controller's preferences; Have a knowledgeable programmer sitting right beside the operator able to make changes to the interface on the fly. This programmer must be very familiar both with the system we are trying to build and the development environment and must also be well versed in HSI science. Once the programmer makes requested changes to the interface, the operator gets to try these changes immediately with the high-fidelity simulation. One of my biggest complaints throughout the design process was the time it took to get from a Power Point concept to a working prototype of the controls. It took most of a year before I could see and use any of the control panels that we discussed in our HSI working

group. This delay was very frustrating. I knew all along we could talk about what we would like to see, but we really would not know what we liked and what we would need until we used it in a realistic situation.

The Operator Vehicle Interface Lab (OVI) within the Human Effectiveness (AFRL/HE) Directorate in the Air Force Research Lab has been working hard over the last few years to put together the hardware, software and network connectivity for "perfect" HSI design environment that I have described. They have been using readily available personal computer (PC) based hardware and software for their system. With their system, they are able to design prototype interfaces for the X-45 program to use within its design process. A slowdown occurs after the contractor receives the OVI prototypes because of antiquated development software that is hosted on very expensive, proprietary hardware. Some of our displays are actually hand coded by programmers using Power Point slides with drawings of what the display should look like. This part of the process is inefficient in both time and schedule. Fortunately, management is learning from the OVI process and our program is moving towards using PC based hardware and software for both design and implementation of our future HSI.

In summary, I have been a part of the design and flight test team for the X-45 system since early in the program. From this experience, I have come to the following conclusions that should help to reduce time and money spent on producing a useable, safe and robust interface for an unmanned bomber system.

- 1) Operators should be included in the design process from day one.
- 2) The design process should allow for immediate changes in the HSI
- 3) The HSI should be integrated with a highly accurate simulation to allow the operator to use in the interface in as real a scenario as possible and as early in the design process as possible.
- 4) The HSI should be user-configurable and allow for multiple modes of feedback and input.

Bottom line: Design and implementation tools that allow real-time prototyping and use with a simulation are essential.